



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide


THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

ORBIT: an optimizing compiler for scheme

 Full text  Pdf (1.38 MB)

Source [Symposium on Compiler Construction](#) [archive](#)
Proceedings of the 1986 SIGPLAN symposium on Compiler construction [table of contents](#)
 Palo Alto, California, United States
 Pages: 219 - 233
 Year of Publication: 1986
 ISSN:0362-1340
[Also published in ...](#)

Authors [David Kranz](#) Yale Univ., New Haven, CT
[Richard Kelsey](#) Yale Univ., New Haven, CT
[Jonathan Rees](#) Massachusetts Institute of Technology, Cambridge, MA
[Paul Hudak](#) Yale Univ., New Haven, CT
[James Philbin](#) Yale Univ., New Haven, CT

Sponsor [SIGPLAN](#): ACM Special Interest Group on Programming Languages

Publisher ACM Press New York, NY, USA

Additional Information: [references](#) [citing](#) [index terms](#) [collaborative colleagues](#) [peer to peer](#)

Tools and Actions: [Discussions](#) [Find similar Articles](#) [Review this Article](#)
[Save this Article to a Binder](#) [Display in BibTex Format](#)

DOI Bookmark: Use this link to bookmark this Article: <http://doi.acm.org/10.1145/12276.13333>
[What is a DOI?](#)

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

- 1 [Harold Abelson , Gerald Jay Sussman , Julie Sussman, Structure and interpretation of computer programs, Massachusetts Institute of Technology, Cambridge, MA, 1984](#)
- 2 [Rodney A. Brooks , Richard P. Gabriel , Guy L. Steele, Jr., An optimizing compiler for lexically scoped LISP, Proceedings of the SIGPLAN '82 symposium on Compiler construction, p.261-275, June 23-25, 1982, Boston, Massachusetts, United States](#)
- 3 Clinger, W., et al. The Revised Revised Report on Scheme, or An UnCommon Lisp. AI Memo \$4~, Massachusetts Institute of Technology, Aug., 1985.
- 4 Ellis, J.R. Bulldog: A Compiler for VLIWArchitecture8. Ph.D. Th., Yale Univ., 1985. available as Research Report YALEU/DCS/RR-364.
- 5 Fisher, J.A. Very Long Word Architectures. YALEU/DCS/RR-253, Yale University, Dec., 1982.



US Patent & Trademark Office

[Subscribe](#) (Full Service) [Register](#) (Limited Service, Free) [Login](#)

Search: ☐ The ACM Digital Library ☒ The Guide



THE GUIDE TO COMPUTING LITERATURE

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Rabbit: A Compiler for Scheme

Source Technical Report: AITR-474
Year of Publication: 1978

Author [Guy L. Steele, Jr.](#)

Publisher Massachusetts Institute of Technology Cambridge, MA, USA

Additional Information: [abstract](#) [citations](#) [collaborative colleagues](#)

Tools and Actions: [Discussions](#) [Find similar Technical Reports](#) [Review this Technical Report](#)
[Save this Technical Report to a Binder](#) [Display in BibTex Format](#)

↑ **ABSTRACT**

We have developed a compiler for the lexically-scoped dialect of LISP known as SCHEME. The compiler knows relatively little about specific data manipulation primitives such as arithmetic operators, but concentrates on general issues of environment and control. Rather than having specialized knowledge about a large variety of control and environment constructs, the compiler handles only a small basis set which reflects the semantics of lambda-calculus. All of the traditional imperative constructs, such as sequencing, assignment, looping, GO TO, as well as many standard LISP constructs such as AND, OR and COND, are expressed as macros in terms of the applicative basis set. A small number of optimization techniques, coupled with the treatment of function calls as GO TO statements, serves to produce code as good as that produced by more traditional compilers.

↑ **CITINGS 3**

[Yukiyoshi Kameyama, Masahito Hasegawa, A sound and complete axiomatization of delimited continuations, ACM SIGPLAN Notices, v.38 n.9, p.177-188, September 2003](#)

[Olivier Danvy, Lasse R. Nielsen, A first-order one-pass CPS transformation, Theoretical Computer Science, v.308 n.1-3, p.239-257, 3 November 2003](#)

↑ **Collaborative Colleagues:**

[Guy L. Steele:](#)

[Paul W. Abrahams](#)
[Ole Agesen](#)
[Eugene Albert](#)
[Robert S. Boyer](#)
[Mark Bromley](#)
[Rodney A. Brooks](#)
[Robert Cartwright](#)
[David Detlefs](#)

[Christine H. Flood](#)
[Daniel P. Friedman](#)
[Richard P. Gabriel](#)
[Alex Garthwaite](#)
[Alexander T. Garthwaite](#)
[Geoffrey S. Goodfellow](#)
[Samuel P. Harbison](#)
[Steven Heller](#)
[W. Daniel Hillis](#)

[Joan D. Lukas](#)
[Paul Martin](#)
[Paul A. Martin](#)
[Daniel D. McCracken](#)
[Tim McNERNEY](#)
[Kenichi Miura](#)
[Mark Moir](#)
[Craig J. Mundie](#)
[David M. R. Park](#)
[Justin R. Rattner](#)

[Edward S. Schneider](#)
[Robert S. Schreiber](#)
[Nir Shavit](#)
[Nir N. Shavit](#)
[Richard M. Stallman](#)
[Gerald Jay Sussman](#)
[Steven Wallach](#)
[Jon L. White](#)

<u>David L. Detlefs</u>	<u>Johan de Kleer</u>	<u>Eric S. Raymond</u>	<u>David S. Wise</u>
<u>Carl W. Diem</u>	<u>Kathleen Knobe</u>	<u>Andreas Reuter</u>	<u>Donald R. Woods</u>
<u>Jon Doyle</u>	<u>Charles H. Koelbel</u>	<u>Larry Rosler</u>	<u>Mary E. Zosel</u>
<u>Raphael R. Finkel</u>	<u>David B. Loveman</u>		
<u>Gerry Fisher</u>			

The ACM Portal is published by the Association for Computing Machinery. Copyright ?2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)